Interconnections and differences between EA and SOA in government ICT development

Ville Seppänen
University of Jyväskylä
ville.seppanen@titu.jyu.fi

Abstract. Disciplines of enterprise architecture (EA) and service-oriented architecture (SOA) share several common goals. Most notably, they both promise improved interoperability and better alignment of business strategy and solutions of ICT. According to one study, 67 percent of governments have established an EA program and many are planning to do so in the near future. It is an interesting notion that several governments are aiming to implement their EA with a means provided by SOA. This paper focuses on evaluating where these two disciplines are interconnected and how established EA can support the implementation of SOA. It was noticed that EA can answers many of the questions that are likely to emerge during the establishment of SOA and, on the other hand, scope of SOA seems to cover the horizontal dimensions of EA frameworks quite well. However, the aspects detailed on the higher vertical levels of EA go beyond the scope of SOA. SOA can provide an organization with tools to better align its ICT with business strategies but as such it does not give a means to formulate and manage the strategic goals. On the other hand, EA does not provide any direct solutions for the problems it can recognize. Therefore, these to disciplines should not be treated as alternatives but instead in parallel.

Introduction

Disciplines of enterprise architecture (EA) and service-oriented architecture (SOA) share several common goals. Most notably, they both promise improved interoperability and better alignment of business strategy and solutions of information and communication technologies (ICT). Several governments around the world have been establishing EA development programs to reduce operational costs, improve efficiency and cross-government interoperability. Almost as many are planning to implemented the renewal of government ICT functions as a means of SOA. The purpose of this paper is to examine what is the actual relation between the disciplines of EA and SOA, and how the one can help the other.

This paper is organized as follows. The next chapter discusses the conception SOA as it is seen today. The third chapter focuses the conception of enterprise architecture and its use in the governmental context. The fourth chapter compares these two and, to find the common ground, puts the special attention on that how they treat the architectural layers of abstraction. The fifth chapter draws an example from the Finnish government EA work by evaluating the support it would give for a service-oriented implementation. The sixth chapter presents the conclusions.

Service-oriented architecture

In a service-oriented architecture, functionality of software applications is represented as services. Characteristic to SOA services is that they are well-defined and self-contained modules that provide standard business functionality. Such services have published interfaces that are described using a standard definition language. Services communicate with each other by requesting the execution of operations to collectively support a business task or process. In order for a service to be self-contained, it must not be dependent on the state or context of other services. (Papazoglou & van den Heuvel, 2007)

The ideas of service-oriented computing are nothing new. For example, Common Object Request Broker Architecture (CORBA) implements self-contained and self-describing application services that are encapsulated with a standardized interface. SOA, however, gained a momentum soon after the emergence of Web services technologies. These open XML-based standards enabled easy and cost effective implementation of technological interoperability regardless on what platforms the interconnecting systems were developed on. At first, SOA was seen merely as another technological solution for implementing distributed computing systems. Thus, the term architecture basically referred to application architecture. Recently, the scope of SOA has expanded. A modern SOA is more about thinking in terms of service orientation and less about implementation. Functionality provided by software applications are abstracted to coarse-grained services that represent the relevant business concepts. New applications, such as business processes, can be created by combining the existing services located in a service repository. As Web services technologies enable standardized platform-independent interactions over the Internet, organizations can easily employ the services their business partners offer and likewise offer their own services to their partners. Recently proposed specifications, such as WS-HumanTask (IBM, 2007a) and BPEL4People (IBM, 2007b), that allow incorporating manual tasks and human interaction into Web services based processes even further the conditions of SOA to become a comprehensive architectural model to design and implement organizational operations and functional structures.

SOA has enabled the introduction of the service-oriented enterprise (Demirkan, Goul & Brown, 2007). A concept of service-oriented enterprise (SOE) has gained an attention of both academia and practitioners. According to Janssen (2008), SOE can be viewed as a "set of capabilities that can be reconfigured to meet changing objectives" and it "moves organizations from application development to focus on the assembly of components and the coordination of those components". A prerequisite for SOE is a widely spread service orientation that covers a great share of organization's software applications and other functional resources. In a sense, a principle according to which all functions in SOA are defined as services (Arsanjani, 2002) is extended into an organizational context in SOE. This conception is well nurtured in the recent literature. Service is the main concept of SOA and identifying the services in an organization is a focal part in creating and optimizing it (Knippel & Skytte, 2007). Khoshafian (2007) depicts three fundamental levels of SOE and the corresponding components of service orientation. The foundational level is that of service oriented architecture (IT service infrastructure), which includes the infrastructure guaranteeing service, quality of service, and enterprise service bus for intra- and inter-enterprise connectivity. The topmost level is the enterprise performance management (service performance). At this level, different departments within an organization, or different organizations, are brought together to realize business goals. Departments (or organizations) participate in a value chain, each adding value towards the ultimate product. As Khoshafian (2007) remarks, SOE assembles services and publishes them as composite applications. The middle layer of SOE is the business process management (service integration) layer. This layer brings the business and IT together. Business process management systems model and execute the interactions among human actors, information systems, and trading partners. (Khoshafian, 2007) The business goals defined at uppermost level of the SOE are here implemented by composing and consuming the services provided by the IT service infrastructure.

According to Arsanjani (2002), successful development projects have been increasingly focusing on larger-grained components providing larger amount of business functionality that is relevant from the larger enterprise perspective. This orientation seems to be reigning in ongoing SOA projects. For example, in their SOA project, British Telecom has been identifying capabilities that have high business relevance. These capabilities, each with between five and 15 operations, are then exposed as services of large granularity (Roberts, 2007). The idea of representing software functionality as coarse-grained services that can be directly mapped to corresponding business functionalities offers a great value for aligning IT with business goals.

Enterprise architecture

Enterprise architecture can be seen as a means for coping with increasing complexity of modern organizations (Zachman, 1987; Shah & Kourdi, 2007). EA identifies and models the functional components of an organization and that how they are interrelated. By this means, EA is not a physical product in itself (Knippel & Skytte, 2007) but rather a process that produces the products (e.g., models and descriptions) that describe the current and the desired structure of an enterprise. It also can – and often does – comprise the governance structures required to further enforce the process so that the desired state can be reached. EA describes how information systems, processes, organizational units, and people in an organization function as a whole (Morganwalp & Sage, 2004). Most of the current EA frameworks used to manage this entirety are based on the pioneering work of Zachman (1987) and feature the viewpoints of business, information, applications, and technology in addition to several levels of abstraction and decision-making. The different levels of decision-making make it possible to manage the organizational complexity and to target different groups of stakeholders. For example, the models that address business executives must be represented in a different manner compared to those addressed to IT staff.

In their enterprise architecture metamodel, Braun and Winter (2005) separate the layers of strategy, organization, and application. The design goal of the strategy layer is the positioning of the organization (business unit) in the value network. According to Braun and Winter (2005), also the specification of product / services and specification of organizational goals are located on the strategy layer. On the organizational layer the design goals are efficiency and effectiveness of product / service innovation, production, and distribution. Specifications of business processes, key performance indicators, organizational structure, and information needs of business and management processes are the most important products of this layer. The application layer creates a link between information system components and business requirements. At this level, applications are representing high-level conceptual constructs that are derived from analyzing activities, information usage and creation, and responsibilities. At the lowest level, there is a layer of software components whose general goal for design is the reuse of software artifacts and respective data structures. (Braun & Winter, 2005)

The enterprise architecture metamodel by Braun and Winter (2005) is in accordance with the layers for software architecture presented by Malan and Bredemeyer (2002). It includes an enterprise scope that consists of one or more domain scopes, which again include application scopes that consist of software component scopes. The corresponding levels of decision-making in the software architecture model are enterprise architecture and top management decisions, domain architect and middle management decisions, application architect decisions, and component owner and developer decisions (Malan & Bredemeyer, 2002; Hirvonen & Pulkkinen, 2004). Hirvonen and Pulkkinen (2004) propose these layers for the EA decision-making in their EA management grid and the similar idea was later implemented in the Finnish government EA method and framework.

Recently, several governments around the world have initiated EA programs. The programs pursue goals such as reduced operational costs, improved efficiency, and cross-government interoperability (Liimatainen et al., 2007). According to Christiansen and Gotze (2007), 67 percent of governments that answered the survey have established an EA program and many are planning to do so in the near future. It is an interesting notion that several governments are aiming to implement their EA with a means provided by SOA (Liimatainen et al., 2007). Although there is some evidence that quite a few of the goals of national EA programs can be achieved just by applying some principles of SOA, such as using well-defined and self- describing services in a loosely coupled manner (e.g., Kalja, Reitsakas & Saard, 2005; Kalja, Kindel, Kivi & Robal, 2007), and from another point of view, implementing a successful SOA can require taking almost as broad perspective as EA does (e.g., Ibrahim & Long, 2007; Linthicum, 2004), it might be advised to take a stance of Schekkerman (2006) towards SOA. He notes that SOA is an architectural style and not architecture itself, or in other words, "SOA has less to do with architecture itself but is a style characteristic of architecture" (Schekkerman, 2006).

Interconnections between EA and SOA

Given the premises presented in the previous sections, the remainder of this paper will focus on evaluating how an established EA can be realized by means of SOA, or from a slightly different perspective, how EA can support the implementation of SOA. To achieve this, the interconnections that exist between the principles of EA and SOA are examined.

As Ibrahim and Long (2007) note, there is a great deal of overlap in concepts, activities, processes, and outcomes of EA and SOA. Knippel and Skytte (2007), for one, notice that originally EA and SOA abstracted away from the technology to grasp the complexity of IS (Zachman, 1987) and to grasp the ever-changing nature of the business (Scheider, 2007), correspondingly. However, as a result of evolution of the disciplines, they now share a common goal that is creation of an agile cost effective business (Knippel & Skytte, 2007).

Despite of that SOA does not separate the different levels of abstraction or decision-making as explicitly as many EA frameworks do, the same idea is still prevailing in the service-oriented approach. In general, the term service orientation implies different set of issues and activities to different audiences (Kontogiannis, Lewis, Smith, Litoiu, Müller, Schuster & Stroulia, 2007). For example, to software engineers SOA is about functional requirements, components, integration techniques, messaging, tools, development environments, and middleware, whereas to business people it is about implementing business strategies, enabling leaner IT departments, facilitating agile process models, and driving new service-delivery processes, and to software-system end-users it is about transparency, flexibility, and ubiquitous access to different services (Kontogiannis et al., 2007).

It becomes evident from the above, the scope of SOA goes well beyond mere software architecture. Whether such a broader conception of SOA should be still referred to as SOA or with terms such as SOE or extended SOA (Papazoglou, 2003; Papazoglou, 2005) instead is a matter of opinion.

The extended SOA (xSOA) model features architectural layers to realize the separation of concerns. Each layer defines a set of constructs, roles, and responsibilities and leans on constructs of its predecessor layer. According to Papazoglou & van den Heuvel (2007), there is a need to separate basic service capabilities provided by the conventional SOA (the bottom layer of xSOA) from more advanced service functionality needed for creating composed services (composition layer), and further, the need to distinguish the functionality of composing services from the management of services (the topmost management layer).

Along with business process modeling techniques and execution languages gaining momentum in the SOA domain a three-layered infrastructure has well established itself in building service-oriented applications. The smallest-granule services or functions of existing applications wrapped as services provide the service building blocks. These are then used to create composition services through orchestration techniques. Finally, at the top-level of the infrastructure hierarchy, business processes execute by using either composite services or individual services. (Khoshafian, 2007) Ibrahim and Long (2007) present a SOA solution stack that combines the layers of atomic and composite services and adds a service components layer below. The service component layer maps to the operational layer containing the representation of existing applications used in an organization.

Ibrahim and Long (2007) mention similarities between EA and SOA domains are that they both address similar architectural domains, are intended to closely align IT with business, use input based on business objectives, and require similar strategies and planning activities. However, according to a report on government enterprise architecture work in 15 countries (Liimatainen et al., 2007), SOA is not generally seen as a complementary to an EA. Rather, it is taken as a means to implement an EA. This is sound since EA itself does not provide any solution models but it rather addresses the strategic and functional requirements of an organization and potentially points out the areas in which these can be implemented. For example, an enterprise architect may recognize a need for application integration on a certain domain. Meanwhile, SOA can provide a (technological) model for implementing this need. However, as Ibrahim and Long (2007) notice, though the SOA approach to integration may prove to be the recommended approach, an enterprise architect should only consider it as one of the approaches EA needs to define and support. By managing the big picture, EA can help an organization to identify the best solution for an individual problem – be it SOA or something else.

On the other hand, should an organization decide to implement their EA by a means of service-oriented technologies, it is important to identify that where and how these two disciplines are interrelated and possibly overlapping. The development of an EA is usually divided into phases of creating the baseline architecture of an organization (the "as-is" view) and the target architecture (the "to-be" view) (Shah & Kourdi, 2007). The baseline architecture models the current structure of an organization, the different layers and existing components. It serves as a starting point that identifies the relationships between the different components as well as areas where there is room for improvement. The target architecture models the structure of an organization as it should be in order to achieve the strategic and operational goals. The architectural roadmaps describe the possible phases of transition between these stages. According to Shah and Kourdi (2007), the roadmaps should "represent the baseline architecture's intermediary alternatives while mitigating the risks and analyzing existing gaps during the shift to the target architecture". SOA comes into consideration while an organization is designing its target architecture and in process of implementing the transition phase.

There exist a few different interpretations about how concept of SOA (or, in more general terms, service orientation) map to the one of EA. Schmelzer and Bloomberg (2006) put SOA in the context of Zachman Framework by placing the very concept of SOA in the intersection of the column "Function" and the row "Logical System Model". This is justified, as SOA definitely started as an application architecture that this intersection represents in the Zachman Framework. However, as SOA has grown out of being merely application architecture, Schmelzer and Bloomberg (2006) note that SOA also represent business processes as compositions of services and expose processes to services. It also affects the way that information is shared and represented (semantic and logical data models) and that how network deals with applications. Schmelzer and Bloomberg (2006) therefore conclude that SOA involves also the eight neighboring squares of "Application Architecture" in the Zachman Framework.

The Institute for Enterprise Architecture Developments (IFEAD) broadens the scope of service orientation even further with regard to EA. The concepts Services Paradigm Adoption, Services Oriented Enterprise, Service Oriented Computing, Services Transition Plan, Governance of Services, Services Oriented Maturity, Choreography of Services, Quality of Services, and Maturity of Services in addition to Service Oriented Architecture are well enough to cover almost all the aspects that EA may cover. (Schekkerman, 2006; Knippel & Skytte, 2007) As Knippel and Skytte (2007) notice, the evolution of SOA has resulted in SOA covering issues that are normally considered as part of EA. Therefore, they argue that EA and SOA should not be treated as separate disciplines but instead they should be brought together.

Finnish government EA framework from the viewpoint of service-oriented implementation

In this section, we will take a look at the Finnish government EA framework and the conditions it provides for enabling service-oriented implementations. The Finnish EA method (Hirvonen, Pulkkinen & Valtonen, 2007) was created during the autumn 2006 and spring 2007 in the Interoperability development program that is a part of the program of government ICT management. The framework of Finnish EA method uses the levels of decision-making that are comparable to the ones of enterprise, domain, and applications that are well established and either directly or indirectly referred to in several architectural frameworks in the fields of software architectures, enterprise architectures, and service-oriented architectures. That said, however, it should be noticed that Finnish EA method was not specifically developed with a service-oriented implementation in view. As a matter of fact, compared to an average government-level EA work (Liimatainen et al., 2007), the work in Finland seems to be lacking all the direct references to the possibilities of SOA.

The Finnish framework in its common form models the government enterprise architecture on the levels state administration, administrative branches, and agencies. The topmost level depicts the issues that describe the government's components en bloc (e.g., the services portfolio of public administration) and that are to be followed and implemented on the subsequent levels (e.g., architectural principles and standards). The middle level collects the architectures of administrative branches, e.g., ministries and cross-branch administrative clusters. The bottom level of the framework is used to draw together the architectures of governmental agencies and offices. Architectural decisions made at this level are supposed to adhere to those of corresponding administrative branch or cluster.

The architectural viewpoints used in the framework are those of business, data, applications, and technology, familiar from several well-known EA frameworks such as FEAF, TOGAF, and E2AF. The viewpoint of business architecture describes the core-business processes that implement government's strategic goals, supporting processes, and other functional resources such as centralized services. Data architecture is used to model essential and strategic information reserves, and related semantics and data models. Data architecture aims at rationalizing the use of information and reducing the duplicate resources. Applications architecture describes the systems that support the execution of business processes and make use of data resource. Technology architecture delineates available technological solutions and standards so that the strategic goals can be effectively achieved.

Thus far the Finnish EA method has been used in pilot projects of two governmental agencies, the Road Administration and the State Treasury. The projects produced agency-level adaptations of the EA framework. The framework adaptations for the agencies showed a resemblance with the architectural separation of layers of enterprise, domains, and applications. The adapted framework for Road Administration features the original architectural viewpoints (business, data, applications, and technology). The framework presents the three levels of decision-making. The uppermost level describes the organization as a whole, from the business-driven and highly abstracted point of view. This level defines the strategies and policies that are reflected on the following layers of the architecture and the concepts that are common to all operational levels of the agency. These contain, for example, applications and services portfolios including the related interface descriptions, central data reserves and vocabularies, standards, and recommended technologies. The next level describes the domains. Domains may represent organizational departments but also other operational units such as services and processes that are extensive enough to be treated as architectural entities. The lowest level of the adapted framework – the systems level – features also the lowest level of abstraction. It describes the domain-related operational processes, systems and reference architectures. Moving down on the layers of the EA framework the level of abstraction decreases and correspondingly the level of technological details increase. At the same time, the focus of models is narrowed down from the strategic goals to the structures implementing them.

Although Linthicum (2004) notes that SOA is a situational thing and its implementation is closely tied to the organization implementing it he has presented the steps for the creation of SOA based on the patterns emerging around the conception of SOA. The fist two steps, "Understand your business objectives and define success" and "Define your problem domain", are well treated by the Finnish EA framework as well as other common EA frameworks. The level of enterprise exclusively focuses on strategic business goals that are pursued in the organization. On the other hand, a comprehensive view that EA provides to an organization helps in identifying the problem domains and the areas that require enhancements. The next step, "Understand all application semantics in your domain", is well covered by the viewpoint of the baseline applications architecture. The services portfolio models of baseline business architecture on the architectural levels of enterprise and domains can respond the questions posed during the step "Understand all services available in your domain". Linthicum's (2004) fifth step, "Understand all information sources and sinks available in your domain" goes into the domain of data architecture as it models the information reserves, vocabularies, and semantics at the upper levels, and the data schemas and structures, at the lowest level. Sixth step, "Understand all processes in your domain", is treated in the business architecture viewpoint by its process models ranging from strategic core-business processes at the enterprise level to the operational processes at the systems level. The step seven advises to "Identify and catalog all interfaces outside of the domain your must leverage". At this phase a traditional EA work may fall short but given the cross-government nature of the governmental EA work all public agencies are presupposed to model their applications and services interfaces in accordance with the standards defined in EA. Therefore, when a government-level EA has been established, the agencies can suppose to find compatible interface descriptions from any other agency their need to cooperate with. In the Finnish EA framework, the interface descriptions are modeled by the viewpoint of applications architecture. The last five steps, "Define new services and information bound to those services", "Define new processes, as well as services and information bound to those processes", "Select your technology set", "Deploy SOA technology", and "Test and evaluate" are omitted here since correspond to the decisions related to the target architecture.

Conclusions

Enterprise architecture and service-oriented architecture share several goals that are common to both disciplines. They both address similar architectural domains, require input that is based on business objectives, provide devices to align business strategies and the use of ICT, pursue agile and cost effective business, and are used to improve organizational flexibility and interoperability. They also require similar strategies and planning activities. There seems to be evidence for that both disciplines in many cases are in favor of devising different levels of abstraction. However, this is where SOA exposes its origins as the application architecture. Whereas levels of abstraction in EA frameworks and models clearly make a difference according to the levels of an organization, decision-making, or viewpoints of different stakeholders, SOA mainly separates the levels of application abstraction, e.g., service components, atomic services, composite services, and business processes. Therefore, EA and SOA cannot be seen as alternatives to each other. However, they do not exclude each other. On the opposite, SOA may provide a great means for implementing the strategies delineated by the EA and, on the other hand, as shown by Estonia's X-Road platform (Kalja et al., 2005; Kalja et al., 2007), a SOA based solution can effectively and affordably grasp the goals that many governments pursue with their EA driven e-government programs.

The scope of SOA seems to cover the horizontal dimensions of EA frameworks quite extensively as it commits on all the typical architectural viewpoints from business to technology. However, the aspects detailed on the higher vertical levels of EA go beyond the scope of SOA. SOA can provide an organization with tools to better align its ICT with business strategies but as such it does not give a means to formulate and manage the strategic goals. On the other hand, EA does not provide any direct solutions for the problems it can recognize. Therefore, these two disciplines should not be treated as alternatives.

By comparing the steps that are recommended to follow while creating a SOA (Linthicum, 2004) with an exemplary EA framework it was noticed that EA can answers many of the questions that are likely to emerge during the establishment of SOA. This was in spite of that the Finnish EA framework used in the comparison was not developed with service-oriented implementation in mind. This finding gives further support to that it is advised to use the approaches of EA and SOA in parallel. EA can greatly contribute to and promote the establishment of SOA. Then again, as they both approach similar problem domains and share several common goals, SOA being more technologically oriented approach can provide a practical means to implement strategic objectives recognized during the EA development work.

Acknowledgments

The author would like to thank the anonymous reviewers for their constructive comments. This study is conducted in the FEAR project (http://www.jyu.fi/titu/fear/in_english). The project is funded by the Finnish Ministry of Finance, IBM, Ixonos, Microsoft, SAS Institute and TietoEnator.

References

Arsanjani, A. (2002). "Introduction to the special issue on developing and integrating enterprise components and services", Communications of the ACM, vol. 45, no. 10, 30-34, October 2002

Braun, C., Winter, R. (2005). "A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform", Desel, J., Frank, U. (Eds.): *Enterprise Modelling and Information Systems Architectures*, Proc. of the Workshop in Klagenfurt, GI-Edition Lecture Notes (LNI), Klagenfurt, 24.10.2005, Gesellschaft für Informatik, Bonn, P-75, 2005, pp. 64-79.

Christiansen, P. E., Gotze, J. (2007). “Trends in Governmental Enterprise Architecture: Reviewing National EA Programs – Part 1”, Journal of Enterprise Architecture, vol. 3, no. 1, 8-18.

Demirkan H., Goul M., Brown G. W. (2007). “Towards the Service-oriented Enterprise”, 40th Annual Internation Conference on System Sciences (HICSS40), Big Island, Hawaii

Hirvonen, A., Pulkkinen, M. (2004). ”A Practical Approach to EA Planning and Management: The EA Management Grid” Abramowicz, W., BIS, (Eds.), Proceedings of 7th International Conference on Business Information Systems. Poznan, Poland, pp. 284–302.

Hirvonen, A., Pulkkinen, M., Valtonen, K. (2007). Selection Criteria for Enterprise Architecture Methods, ECIME Montpellier, France.

IBM (2007a). “Web Services Human Task (WS-HumanTask), Version 1.0, available online at http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf, referred to 2.4.2008

IBM (2007b). “WS-BPEL Extension for People (BPEL4People), version 1.0”, available online at http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf referred to 2.4.2008

Ibrahim, M., Long, G. (2007). ”Service-Oriented Architecture and Enterprise Architecture, Part 2: Similarities and Differences”, available online at

Janssen, M. (2008). “Exploring the Service-Oriented Enterprise: Drawing Lessons from a Case Study”, Proceedings of the 41st Annual Hawaii International Conference on System Sciences

Kalja, A., Reitsakas A., Saard N. (2005). ”eGovernment in Estonia: Best practices”, In Proceedings of PICMET’05 (Portland, USA, July 31- Aug. 4)

Kalja, A., Kindel K., Kivi R., Robal T. (2007). “eGovernment srvices: How to develop them, how to manage them?”, In Proceedings of PICMET’07 (Portland, USA, August 2007)

Khoshafian, S. (2007.) *Service Oriented Enterprises*, Auerbach Publications, Taylor & Francis Group, Boca Ranton, New York

Knippel, R., Skytte, B. (2007). “Service Oriented Enterprise Architecture: An Opportunity of Evolution”, Journal of Enterprise Architecture, Vol. 3, No. 3, pp. 17-25, August 2007

Kontogiannis, K., Lewis, G. A., Smith, D. B., Litoiu, M., Müller, H., Schuster, S., Stroulia, E. (2007). ”The Landscape of Service-Oriented Systems: A Research Perspective”, International Workshop on Systems Development in SOA Environments (SDSOA’07)5

Liimatainen, K., Hoffmann, M., Heikkilä, J. (2007). “Overview of Enterprise Architecture work in 15 countries”, Ministry of Finance, State IT Management Unit, Research reports 6b/2007. http://www.vm.fi/vm/en/04_publications_and_documents/01_publications/04_public_management/20071102Overvi/FEAR_ENGLANTI_kokonaan.pdf

Linthicum, D. (2004). “12 Steps to Implementing a Service Oriented Architecture”, Grend Central Communications, October 2004

Malan, R., Bredemeyer, D. (2002). ”Less is More with Minimalist Architecture”, IT Pro, September/October, 2002, pp. 46-48

Morganwalp, J. M., Sage, A. P. (2004). “Enterprise Architecture Measures of Effectiveness”, International Journal of Technology, Policy and Management, vol. 4, no. 1, pp. 81-94.

Papazoglou, M. P. (2003). ”Service-oriented Computing: Concepts, Characteristics and Directions”, Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE 2003

Papazoglou, M. P. (2005). ”Extending the Service Oriented Architecture”, Business Integration Journal, February 2005, pp. 18-21

Papazoglou, M. P., van den Heuvel, W. (2007). “Service oriented architectures: approaches, technologies and research issues”, The VLDB Journal, Vol. 16, No. 3, July 2007

Roberts, P. F. (2007). “BT plans to be 100 percent SOA by ‘09”, InfoWorld, May 16, available online at http://www.infoworld.com/article/07/05/16/BT_plans_to_be_100_percent_1.html

Schekkerman, J. (2006). “Enterprise Architecture & Service Oriented Architecture (SOA)”, available online at http://www.enterprise-architecture.info/Images/Services%20Oriented%20Enterprise/EA_Service-Oriented-Architecture1.htm

Schmelzer, R., Bloomberg, J. (2006). ”SOA and the Zachman Framework”, ZapThink, Jun 15. 2006

Shah, H. and Kourdi, E.K. (2007). “Frameworks for Enterprise Architecture”, IT Professional, September/October 2007

Zachman, J. A. (1987). A Framework for Information Systems Architecture. IBM Systems Journal, vol. 26, no.3.